Diffusion Maps

Ishan Bhatt*

November 2021

Contents

1 Introduction

Diffusion mapping is a non-linear manifold learning technique introduced by Coifman and Lafon in the mid-2000s [2], first developed in Lafon's dissertation in 2004 [3]. Through a focus on learning local geometry, diffusion maps obtain a global description of the data [4]. This method is related to other kernel eigenmap techniques such as Laplacian Eigenmaps and Hessian Eigenmaps. The non-linearity of diffusion maps provides this technique an edge over linear techniques such as principal component analysis (PCA) or classical multi-dimensional scaling (MDS) in terms of learning the underlying manifold. Additionally, the local nature of diffusion maps preserves local structures, discarding distances between two elements that are very far apart. This property is superior to linear techniques that attempt to preserve distant distances because, in high dimensions, close distances often provide more information [4].

Beginning with the assumption that some data $\vec{y_1} \dots \vec{y_N} \in \mathbb{R}^d$ lie on a Riemannian manifold, diffusion maps attempt to recover lower dimensional representations $\vec{x_1} \dots \vec{x_N} \in \mathbb{R}^s$, with $s << d$. We define the kernel as a

* ishanbhatt@college.harvard.edu, code for figures available at: https://github.com/ishanbhatt42/stat185-final-paper

notion of similarity or *connectivity*, $k(\vec{y_i}, \vec{y_j})$, between two points $\vec{y_i}, \vec{y_j}$ that is proportional to the probability of a random walk transitioning from $\vec{y_i}$ to $\vec{y_j}$ in one step. One can choose between several functional forms of the kernel. After normalization of the kernel, these values can be used to construct a matrix containing the probabilities of transition from one point to another. We are then able to use these probabilities to define a Markov chain on our data.

Running this chain forward in t steps reveals the probabilities of moving from $\vec{y_i}$ to $\vec{y_j}$ in t steps. We then introduce the Diffusion Distance as a metric depending on t and the transition matrix – intuitively, the Diffusion Distance captures the distance between two points by integrating the probability of all paths of length t, which effectively captures paths along the manifold itself.

To reduce dimensionality, we map the data into a lower dimensional space in which the Euclidean distance approximates the Diffusion Distance. The eigenvectors of the Markov matrix generate lower dimensional representations of the data. To reduce dimensionality from d dimensions to s dimensions, we use the first s eigenvectors of the Markov matrix (in order of decreasing eigenvalue) to approximate lower dimension representations of the data. To derive clusters, we can observe groupings in our lower dimensional data since the method emphasizes preserving local geometry.

An advantage of diffusion maps is that we are not left with a single lower dimensional representation but rather a family of diffusion maps through iterating the Markov chain multiple times [1]. That is, we can modify the connectivity definition by instead specifying the probability of transitioning from $\vec{y_i}$ to $\vec{y_j}$ in t steps, modifying the scale at which we define our connectivity.

The most significant drawback of this method is that it it is sensitive to both the choice of kernel and the choice of t (iterations of the Markov chain). While the choice of the kernel and t allows us to explore clusters in different scales [4], the optimal tuning of these parameters often presents difficulty in accurately employing diffusion maps.

I first introduce relevant notation and definitions, mathematically explicating the broad strokes of the technique summarized in the introduction. I then review main result of the technique – that lower dimensional Euclidean distances between mapped points approximate the Diffusion Distances in high dimensions. I review the strenghts and weaknesses of diffusion maps and conclude by demonstrating the technique on the several data sets and comparing it to the performance of other classic clustering and manifold learning techniques.

2 Method

2.1 Markov Chains and their Connection to High Dimensional Geometry

Drawing from [4] and [2], I give an overview of the technique of diffusion mapping. Begin with a set of data points $\mathcal{Y} = \{Y_1, Y_2, \dots Y_N\}^T$ where $Y_i \in \mathbb{R}^d$ and a random walk on $\mathcal{Y}$. We assume the data resides on a lower dimensional manifold of some kind. We define the connectivity p of Y_1, Y_2 as the probability of jumping from Y_1 to Y_2 in one step of a random walk. Analogously, $p_t(Y_1, Y_2)$ is defined as the probability of doing so in t steps, where $t \in \mathbb{N}$. It is common to relate this connectivity proportionally to a kernel, of which there can be many functional forms. That is,

$$p(Y_1, Y_2) \propto k(Y_1, Y_2)$$

Generally, the kernel is a function $k : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ that is non-negative and symmetric[1]. A common functional form is the Gaussian kernel, defined as

$$k(Y_1, Y_2) = \exp(-\frac{||Y_1 - Y_2||^2}{\epsilon})$$

[1] in the sense that $k(Y_i, Y_j) = k(Y_j, Y_i)$

where ϵ is a parameter that allows us to choose the relative scale of the kernel. A crucial characteristic of the kernel is that it quickly vanishes to zero when points are not extremely close to each other. Thus, only points that are very close to each other are given positive transition probability. This characteristic is critical in the power of diffusion maps to accurately uncover true geometric structures underlying the data. To see this, we must first normalize the kernel. Define d_{Y_i} as the constant such that

$$\frac{1}{d_{Y_i}} \sum_{y \in \mathcal{Y}} k(Y_i, y) = 1$$

Let $\mathcal{M}$ be the Markov matrix, such that $\mathcal{M}_{ij} = p(Y_i, Y_j) = \frac{1}{d_{Y_i}} k(Y_i, Y_j)$. By construction, $\mathcal{M}_{ij}$ is the probability of moving from data point Y_i to Y_j in one step of the random walk.

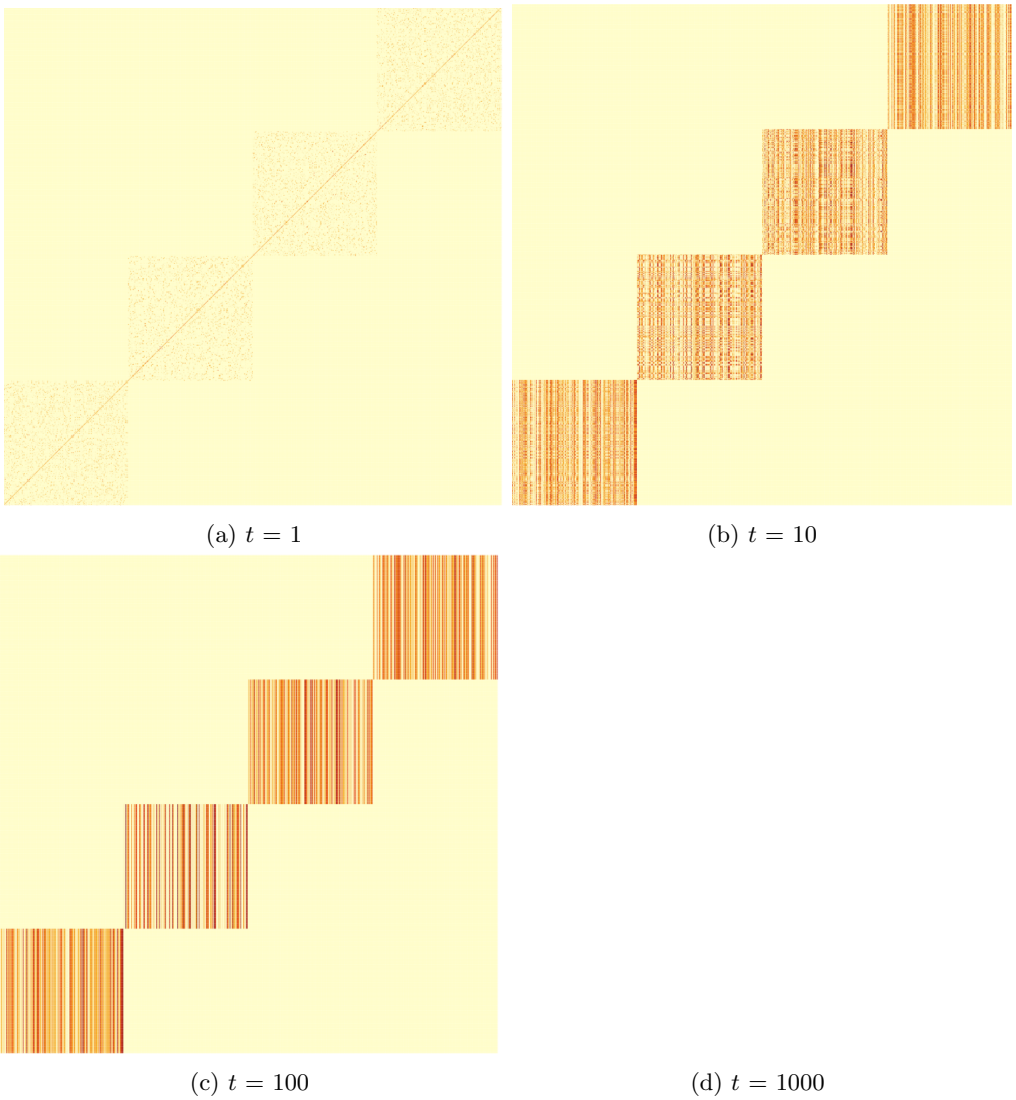
Thus far, we have used a kernel to define a Markov chain on the data with transition matrix $\mathcal{M}$. A natural question the reader may ask is why the construction of $\mathcal{M}$ contains information about the geometry of $\mathcal{Y}$. In fact, based on the observation about the vanishing nature of the kernel for distant points, $\mathcal{M}$ only contains extremely local geometric information.

The key is to take powers of $\mathcal{M}$. Intuitively, the resultant matrix $\mathcal{M}^t_{ij}$ represents the probability of moving from point Y_i to Y_j in t steps – ie $\mathcal{M}^t_{ij} = p_t(Y_i, Y_j)$. Crucially, since the initial kernel assigned non-negligible probabilities to only close points, $\mathcal{M}^t$ contains paths between Y_j and Y_j made up of jumps along points that are "close to each other" on the manifold. Thus, the paths of length t contained in $\mathcal{M}^t$ all lie on the manifold of interest. Thus, the initial local geometric information contained in $\mathcal{M}$ is integrated to represent a global geometry in $\mathcal{M}^t$ as the chain runs *forward in time*.

This also has a connection to the clustering ability of diffusion maps, explored in the following example. Replicating and extending on an example contained in [2], I generate a set $\mathcal{Y}$ of 1000 points in the plane, with a cluster in each quadrant, plotted below. The data points are constructively ordered such that the first 250 are in quadrant I, the next in II, the next in III, and points 750-1000 are in quadrant IV.

A quarter of the points are generated in each quadrant. Using the Gaussian kernel and $\epsilon = 4$, I compute the Markov matrix $\mathcal{M}$ for $t = 1, 10, 100, 1000$. Below, I plot the original data and a heatmap of $\mathcal{M}^t$ at each time stage. The effect of discovering global geometry can be observed graphically.

Figure 1: Quadrant Data and t



(a) $t = 1$ (b) $t = 10$

(c) $t = 100$ (d) $t = 1000$

This simple figure illustrates the learning process of $\mathcal{M}^t$. Observe that at $t = 1$, the matrix gives strong connectivity between points and themselves. It lightly picks up on the four clusters. As the chain runs at $t = 10$ and $t = 100$, the local connectivity is integrating to reveal all four clusters. The reader may think that the $t = 1000$ heatmap is a graphical error of some kind – it is not. At $t = 1000$, the chain functionally has reached everywhere and thus no longer picks up on clusters in the data. At a certain t, a region or cluster can be interpreted as an area of the data the chain is very unlikely to leave and thus has a high probability of staying within points in that region. Hence, the four darker squares that are revealed in $t = 1, 10, 100$. [2].

Given that $\mathcal{M}^t$ contains information about paths of length t on the manifold, it is natural to use P^t to create a measure of distance on the manifold. The Diffusion Distance D_t between two points Y_i and Y_j is defined as [4]:

$$D_t(Y_i, Y_j) = \sum_{\mu \in \mathcal{Y}} |p_t(Y_i, \mu) - p_t(Y_j, \mu)|^2 = \sum_k |\mathcal{M}^t_{ik} - \mathcal{M}^t_{kj}|$$

Intuitively, this distance will be small if two points are well share similar levels of connectivity with points

in $\mathcal{Y}$. That is, if there are a large number of similarly-sized paths from Y_i to Y_j [2]. Lastly, note that this set up yields a family of diffusion distances, indexed by a parameter t, which is a significant advantage of diffusion maps. The Diffusion Distance is, therefore, a robust measure of distance on the manifold.

2.2 Spectral Analysis of $\mathcal{M}$

This section will exploit spectral properties of $\mathcal{M}$ in order to reduce dimension. However, I first begin with a motivating construction from [4]. Consider the mapping:

$$X_i \equiv \begin{bmatrix} p_t(Y_i, Y_1) \\ p_t(Y_i, Y_2) \\ \cdots \\ p_t(Y_i, Y_N) \end{bmatrix}$$

Then, it is immediate that the Euclidean distance between X_i and X_j is the Diffusion Distance between Y_i and Y_j, since

$$||X_i - X_j||^2 = \sum_{k=1}^{N} |p_t(X_i, X_k) - p_t(X_j, X_k)|^2 = D_t(Y_i, Y_j)$$

However, little useful dimension reduction has occurred here – our mapped data X_i is in $\mathbb{R}^N$. Yet, the principle result is hinted at – that Euclidean distances of mapped vectors can approximate the Diffusion Distance.

The main result of diffusion maps is that the first n eigenvectors of the Markov matrix (in order of decreasing eigenvalue) create mapped data such that the *Euclidean distance between the mapped data approximates the Diffusion Distances* between the original data. Before diving into this result, a brief note on the spectral theory employed.

Generally, the spectral theory of $\mathcal{M}$ may not be guaranteed, but under certain technical conditions in our construction omitted in this paper but discussed at length by Coifman and Lafon [2], we do in fact have a spectral decomposition. Let $\mathcal{M}$ have n eigenvectors. Let ϕ_i be the ith eigenvector of $\mathcal{M}$, with λ_i as its corresponding eigenvalue. Observe that the eigenvectors of $\mathcal{M}^t$ are still ϕ_i but have eigenvalues λ_i^t [2]. Assume that the eigenvectors are ordered in terms of largest eigenvalues.

To create a lower dimensional mapping, fixing n as the number of eigenvectors of choice and t as the global scale of choice, we define

$$\Psi_i^{(n,t)} \equiv \begin{bmatrix} \lambda_1^t \phi_1(i) \\ \lambda_2^t \phi_2(i) \\ \cdots \\ \lambda_n^t \phi_n(i) \end{bmatrix}$$

This embeds the data $\mathcal{Y}$ into $\mathbb{R}^n$, reducing the dimensionality of the data. Choosing the first n dominant eigenvectors ensures that

$$||\Psi_i^{(n,t)} - \Psi_j^{(n,t)}|| \approx D_t(Y_i, Y_j)$$

which means we have low-dimensional data $\Psi_i^{(n,t)}$ whose Euclidean distances are approximate the Diffusion Distances of Y_i, which in turn approximate distances between the points of $\mathcal{Y}$ on the manifold. This result is not obvious, and the proof for it in the setting defined above is provided in the appendix of [4]. A different sketch for the result is provided in the next section, albeit in a slightly different setting.

[2] This follows immediately from the definitions as $\mathcal{M}^t \phi_i = \mathcal{M}^{t-1} \mathcal{M} \phi_i = \mathcal{M}^{t-1} \lambda_i \phi_i$. Iterating this yields the result.

2.3 A Small Measure of Measure

Thus far, this paper has focused on spectral decomposition of a Markov matrix using eigenvectors and eigenvalues. However, there is a more general setting for diffusion maps that draws on results from measure theory and functional analysis that more closely follows the work in Lafon's dissertation [3]. The definitions are nearly identical, but are slightly tweaked in this measure-theoretic setting. With this in mind, I provide a proof [3] of the relation between Euclidean distances of diffusion maps and the Diffusion Distance.

1. Let $(\mathcal{Y}, \mathcal{F}, \sigma)$ be a measure space. Let $k : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ be a symmetric, non-negative kernel. Use the same construction for $p(Y_i, Y_j)$ as before. Crucially, note that the normalized $p(Y_i, Y_j)$ is no longer guaranteed to be symmetric.

2. With d_{Y_i} defined as above, define $\pi(Y_i) = \frac{d_{Y_i}}{\sum_{z \in \mathcal{Y}} d_z}$, following [2].

Theorem 1. The diffusion distance, defined below, is equivalent to

$$D_t(Y_i, Y_j) \equiv \int_{\mathcal{Y}} |p_t(Y_i, \mu) - p_t(Y_j, \mu)|^2 d\frac{\sigma(\mu)}{\pi(\mu)} = \sum_{l \geq 1} \lambda_l^{2t} (\psi_l(Y_j) - \psi_l(Y_j))^2$$

Proof. If $p(Y_i, Y_j)$ was still a symmetric operator, our job would be easier. However, it is not and thus we must define

$$a(Y_i, Y_j) = \frac{\pi(Y_i)}{\pi(Y_j)} p(Y_{i,j})$$

which is symmetric [2]. A few other technical conditions also covered in [2] and [1] must also hold. Then, we have a spectral theory on $a(Y_i, Y_j)$ that implies

$$a(Y_i, Y_j) = \sum_{l \geq 0} \lambda_l \phi_l(Y_i) \phi_l(Y_j)$$

where $\phi_l(\cdot)$ is an eigenfunction with corresponding non-negative eigenvalue λ_l. Crucially, $\{\phi_l\}_{l \geq 0}$ are an orthonormal basis of the function space $L^2(\mathcal{Y}, d\sigma)$. This will matter because our diffusion distance is a square integral. Observe that from the construction of a,

$$\begin{aligned} p(Y_i, Y_j) &= \frac{\pi(Y_j)}{\pi(Y_i)} a(Y_i, Y_j) \\ &= \frac{\pi(Y_j)}{\pi(Y_i)} \sum_{l \geq 0} \lambda_l \phi_l(Y_i) \phi_l(Y_j) \\ &= \sum_{l \geq 0} \lambda_l \frac{\phi_l(Y_i)}{\pi(Y_i)} \phi_l(Y_j) \pi(Y_j) \end{aligned}$$

Defining $\psi_l(Y_i) = \frac{\phi(Y_i)}{\pi(Y_i)}$ and $\varphi_l(Y_j) = \phi(Y_j)\pi(Y_j)$ yields

$$p(Y_i, Y_j) = \sum_{l \geq 0} \lambda_l \psi_l(Y_i) \varphi_l(Y_j)$$

and analogously yields

$$p_t(Y_i, Y_j) = \sum_{l \geq 0} \lambda_l^t \psi_l(Y_i) \varphi_l(Y_j)$$

[3] This proof is not original and is adapted from the appendices of [2] and [3]. Some of the later portions are original attempts to expand upon and clarify the arguments in the original paper.

Finally, going back to our Diffusion Distance, we have

$$\begin{aligned}
\int_{\mathcal{Y}} |p_t(Y_i,\mu) - p_t(Y_j,\mu)|^2 d\frac{\sigma(\mu)}{\pi(\mu)} &= \int_{\mathcal{Y}} |\sum_{l\geq 0} \lambda_l^t \psi_l(Y_i)\varphi_l(\mu) - \sum_{l\geq 0} \lambda_l^t \psi_l(Y_j)\varphi_l(\mu)|^2 d\frac{\sigma(\mu)}{\pi(\mu)} \\
&= \int_{\mathcal{Y}} |\sum_{l\geq 0} \varphi_l(\mu)(\lambda_l^t \psi_l(Y_i) - \lambda_l^t \psi_l(Y_j))|^2 d\frac{\sigma(\mu)}{\pi(\mu)} \\
&= \int_{\mathcal{Y}} |\sum_{l\geq 0} \phi(\mu)(\lambda_l^t \psi_l(Y_i) - \lambda_l^t \psi_l(Y_j))|^2 d\sigma(\mu) \\
&= \sum_{l\geq 0} |\lambda_l^t \psi_l(Y_i) - \lambda_l^t \psi_l(Y_j)|^2 \\
&= \sum_{l\geq 0} \lambda_l^{2t} (\psi_l(Y_i) - \lambda_l^t \psi_l(Y_j))^2
\end{aligned}$$

where the second to last line follows from orthonormal basis properties discussed in the appendix of [2]. □

3 Discussion

Perhaps the most important takeaway from the theoretical section of the paper is that since Diffusion Distance can be approximated using the eigenvectors and eigenvalues of $\mathcal{M}$, the technique is relatively computationally inexpensive because one may directly begin the spectral decomposition after constructing $\mathcal{M}$ and will not have to worry about computing Diffusion Distance from the definition.

In addition, there are several advantages to diffusion maps.

First, the technique is non-linear. When dealing with high-dimensional manifolds, particularly very curvy ones, linear techniques often fail to capture distance along the manifold. This is largely because their measures for distance along the manifold are vulnerable to the data points existing in close Euclidean distance and/or the techniques do not discard information relating to data points that are somewhat far apart. Diffusion maps solve both problems, by relying on kernels that go to zero extremely quickly for distant data points and averaging along all paths of length t in order to accurately capture the shape of the high dimensional manifold.

Second, the technique begins with local geometry, and then integrates up to global geometry. The main idea of taking powers of $\mathcal{M}$ allows the technique to begin with the most relevant information – which points are extremely similar – and integrate up to global geometry. This also allows the user inexpensively tune the scale at which they want to view the map.

Third, the diffusion distance as measure of distance on a manifold is robust to noise perturbation [3]. The intuition behind why is that the diffusion distance averages over all possible paths of length t between two points. Thus, misleadingly small Euclidean distances between two points would not affect Diffusion Distance in the same way it might affect geodesic distance.

Fourth, there are theoretical guarantees with regard to choosing some k number of eigenvectors an a notion of an error bound δ. Given an accuracy δ, it can be shown that retaining the n eigenvalues that, when raised to the tth power, exceed a threshold depending on δ, the Euclidean distances in the mapping will approximate the Diffusion Distances with accuracy δ [3].

However, there are two main drawbacks.

First, there are multiple parameters to tune. The technique is sensitive to the choice of t, ϵ, and the kernel functional form itself. While these do create flexibility and adaptability, there are two dangers present. One,

the user may get vastly different results depending on their parameterization. Second, the user may run the risk of overfitting to their data given the choice of the functional form of kernel.

Second, the technique requires dense sampling in order to be effective. Given that the technique begins with local geometry, it is particularly vulnerable to sparse sampling. It can be shown that given a particular error tolerance, the sample must grow faster than a particular technical bound the reader may find in [2].

4 Data Implementation

4.1 Demonstrating the Advantages of Diffusion Maps

I generate a folded washer in three dimensions, made up of $N = 10,000$ data points. I use the numeric code from class, but I modify it to add standard normal noise to each element of each point. I compare the one-dimensional and two-dimensional representations constructed by diffusion maps and classical multidimensional scaling. I first print the folded washer below for reference. A crucial observation is that the colors of the points vary along the non-linear curve of the washer.

Figure 2: Folded Washer of 10000 Data Points

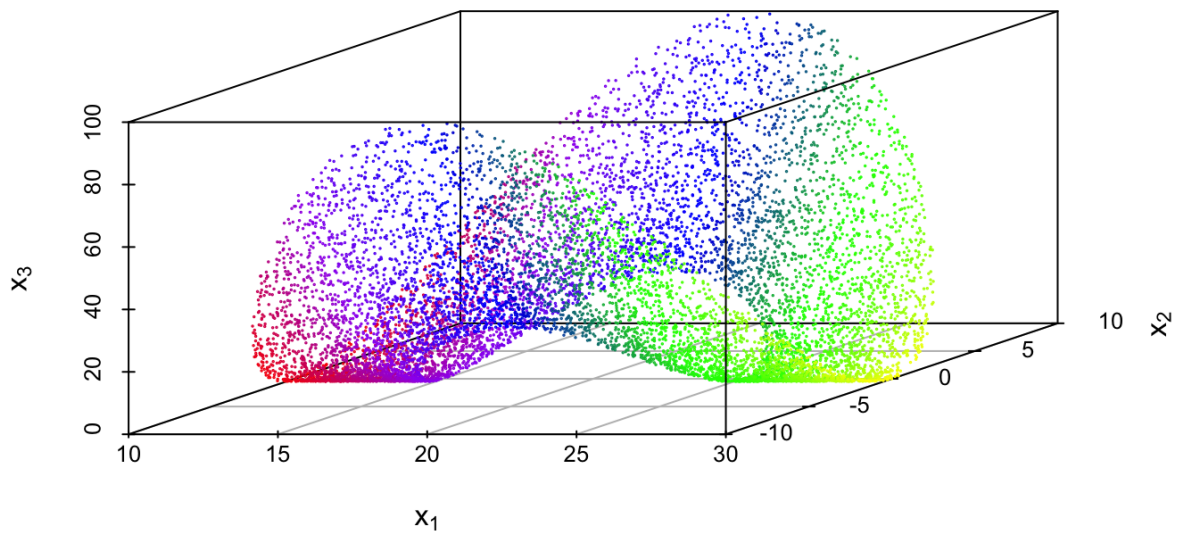
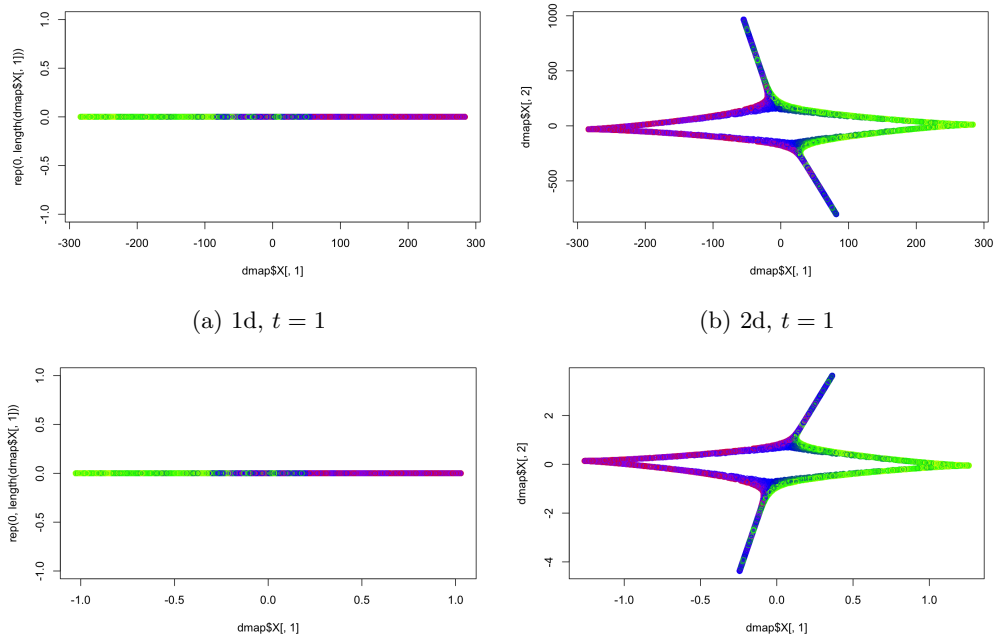


Figure 3 and Figure 4 plot the one and two dimensional representations outputted by diffusion maps and classical MDS. Diffusion maps present fairly reasonable one and two dimensional representations, with their two dimensional representations slightly off of the original circular shape of the washer. Of course, classical MDS presents the best 2d representation. However, the key takeaway and the reason for the inclusion of these figures is that classical MDS in the one dimensional case is unable to capture the colors in the folded

washer. The one-dimensional representation is largely unable to meaningfully preserve cluster order due to its non-linear nature. However, diffusion maps are able to meaningfully preserve those clusters in their one dimensional representations.

Figure 3: Diffusion Maps of Folded Washer



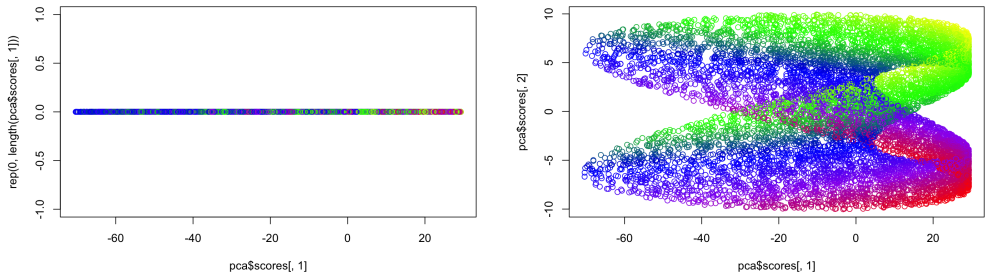
(a) 1d, $t = 1$

(b) 2d, $t = 1$

(c) 1d, $t = 5$

(d) 2d, $t = 5$

Figure 4: Classical MDS of Folded Washer

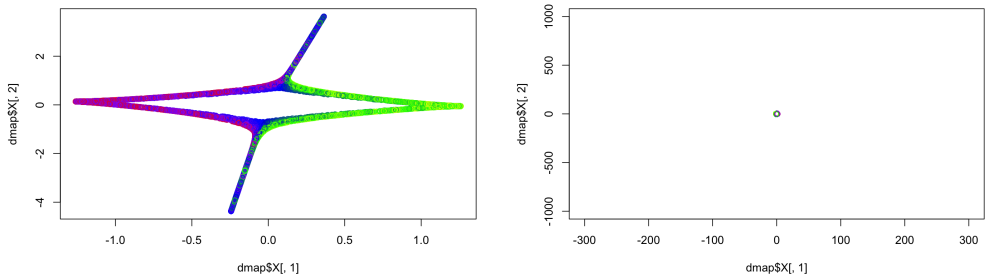


(a) 1d

(b) 2d

It may appear at first glance that in Figure 4, the $t = 5$ case looks extremely similar to $t = 1$. This is not the case if one notices the axes. In the $t = 5$ case, a much more global approach is taken and points are more likely to pushed closer together. Figure 5 plots the two dimensional $t = 5$ case with the automatic axes generated next to the same representation but with the axes from the $t = 1$ case. Again, the effect of t on the output of our diffusion map is vissually immediate.

Figure 5: Comparison of Scale at $t = 1$ and $t = 5$



(a) 2d, $t = 1$

(b) 2d, $t = 5$ with rescale

4.2 Diffusion Maps in Practice

The reader might consider the following practical notes on implementing diffusion maps using the R package `diffusionMap` [5], the package used in this paper.

First, the main function `diffuse` has an option `eps.val = epsilon.compute()` that optimally selects the ϵ parameter in the Gaussian kernel. The reader is free to manually fix their own value, but the functionality to let the package tune the parameter exists.

Second, for fixed values of t, one can plot the eigenvalues in decreasing order akin to a scree plot. If there exists an elbow at a certain k, it means those first k eigenvalues have captured the vast majority of approximation of the diffusion distance. If one allows t to grow, one will notice that it takes fewer eigenvalues to capture the Diffusion Distance as the process captures more of the global structure. This visual process may aid in tuning the t and n parameters.

References

[1] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the National Academy of Sciences, 102(21):7426–7431, 2005. Publisher: National Academy of Sciences _eprint: https://www.pnas.org/content/102/21/7426.full.pdf.

[2] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. Applied and Computational Harmonic Analysis, 21(1):5–30, 2006.

[3] S.S. Lafon. Diffusion Maps and Geometric Harmonics. Yale University, 2004.

[4] J Porte, Ben Herbst, Willy Hereman, and Stéfan van der Walt. An Introduction to Diffusion Maps. November 2008.

[5] Richards, Joseph and Cannoodt, Robrecht. diffusionMap.